

EE318061326US

**A METHOD AND APPARATUS FOR MAINTAINING STATE INFORMATION  
FOR WEB PAGES USING A DIRECTORY SERVER**

5

**1. Field of the Invention:**

The present invention is directed to a method and apparatus for maintaining state information for Web pages using a directory server.

10

**2. Background of the Invention:**

The Internet is fast becoming the most prevalent tool through which goods and services are marketed to customers. The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from the sending network to the protocols used by the receiving network (with packets if necessary). When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty. Many federal, state, and local government agencies are also employing Internet sites for informational purposes, particularly agencies which must interact with virtually all segments of society such as the Internal Revenue Service and secretaries of state. Providing informational guides and/or searchable databases of online public

0350131.041400

Docket No. AUS000078US1

records may reduce operating costs. Further, the Internet is becoming increasingly popular as a medium for commercial transactions.

Currently, the most commonly employed method of transferring data over the Internet is to employ the World Wide Web environment, also called simply "the Web". Other Internet resources exist for transferring information, such as File Transfer Protocol (FTP) and Gopher, but have not achieved the popularity of the Web. In the Web environment, servers and clients effect data transaction using the Hypertext Transfer Protocol (HTTP), a known protocol for handling the transfer of various data files (e.g., text, still graphic images, audio, motion video, etc.). The information in various data files is formatted for presentation to a user by a standard page description language, the Hypertext Markup Language (HTML). In addition to basic presentation formatting, HTML allows developers to specify "links" to other Web resources identified by a Uniform Resource Locator (URL). A URL is a special syntax identifier defining a communications path to specific information.

Each logical block of information accessible to a client, called a "page" or a "Web page", is identified by a URL. The URL provides a universal, consistent method for finding and accessing this information, not necessarily for the user, but mostly for the user's Web "browser." A browser is a program capable of submitting a request for information identified by an identifier, such as, for example, a Universal Resource Locator (URL). A user may enter a domain name through a Graphical User Interface (GUI) for the browser to access a source of content. The domain name is automatically converted to the Internet Protocol (IP) address by a Domain Name System

0950131-041400

(DNS), which is a service that translates the symbolic name entered by the user into an IP address by looking up the domain name in a database.

The Internet also is widely used to transfer applications to users using browsers. With respect to commerce on the Web, individual consumers and business use the Web to purchase various goods and services. In offering goods and services, some companies offer goods and services solely on the Web while others use the Web to extend their reach.

In marketing goods and services, often companies require that the users enter personal information for their records and to effectuate a sale of goods or services. Such information is usually entered via a Web page having fields denoted by HTML tags, such as the <FORM> tag.

**Figure 1** is an example Web page having fields in which a user may enter personal information for use by the Web server. As shown in **Figure 1**, the Web page has fields for entry of the user's name and address. Once the user enters the information in the required fields, the information is sent to the Web page server for processing. The information transmission to the Web page server is accomplished when the user sends a "transmit" command via the user's Web browser. This may be done, for example, by selecting a virtual "enter" button or "next" button on the Web page display.

The ability of Web servers to maintain state information for a Web page is very limited in that state information typically can only be passed from one Web page to the next immediate Web page. Maintaining state information means remembering information while the user moves from Web page to Web page within a Web site. If a

0950191-041400

Docket No. AUS000078US1

user moves beyond the next Web page to another Web page in the Web site, the state information for the first Web page will be lost. Thus, the user is required to reenter all of the information previously entered.

5        There are some possible ways of maintaining state information for longer periods of time. The state information may be stored as cookies on the user's personal computer. Cookies are data created by a Web server that provides a way for a Web site to keep track  
10 of a user's patterns and preferences and, with the cooperation of a Web browser, to store them on the user's own hard disk.

      The cookies contain a range of URLs (addresses) for which they are valid. When the browser encounters those  
15 URLs again, it sends those specific cookies to the Web server. For example, if a user's ID were stored as a cookie, it would save that person from typing in the same information all over again when accessing that service for the second and subsequent time.

20        Cookies, however, have the drawbacks that cookies are only valid for certain dates, i.e. cookies expire. If a cookie expires or is otherwise disabled, such as through a setting in the Web browser, the information stored in the cookie cannot be used.

25        In addition, cookies are limited in their size and are not valid across Web browsers. In other words, a cookie that is created using Netscape™ will not be valid for use with Microsoft Internet Explorer™. The most significant drawback is that cookies are stored in an  
30 unencrypted format. Thus, sensitive information, such as credit card numbers, social security numbers, and the like, which are typical information provided during Internet sales, would not be secure when stored in a

09550181-041400

cookie.

Another option is to encode the state information in the URL. The URL is an address that defines the route to a file on the Web or any other Internet facility. URLs are typed into the browser to access Web pages, and URLs are embedded within the pages themselves to provide the hypertext links to other pages.

For example, a URL such as "www.techweb.com/encyclopedia/defineterm?term=URL" may be utilized to temporarily store Web page state information. The term "defineterm?term=" provides the attribute that is stored in the URL with "URL" being the user entered state attribute value. The drawback to using this approach is that the state information is maintained only temporarily and is lost when the user moves to another Web page.

A still further option is to store the Web page state information on the Web server. This requires that each Web server provide sufficient storage capacity to store Web page state information for each user of each Web page. Additionally, it requires the Web server to have a Common Gateway Interface (CGI) script for interfacing with a Web server database. A CGI is a small program written in a script language, such as Perl, that functions as the glue between HTML Web pages and other programs on the Web server. For example, a CGI script allows search data entered on a Web page to be sent to a database management system (DBMS). It also formats the results of that search onto an HTML Web page, which is sent back to the user.

Thus, storing the Web page state information on the Web server greatly increases the complexity, storage requirements, and ultimately the cost of a Web server. Additionally, the client applications on the user's

0950181.041400

Docket No. AUS000078US1

computer cannot maintain control over the use of the Web page state information or utilize this information with other Web pages.

- 5 Therefore, it would be beneficial to have a method and apparatus for maintaining state information for Web pages that is secure and does not require a Web server for storage of the Web page state information.

09550181-04-1400

**SUMMARY OF THE INVENTION**

5 The present invention is directed to an apparatus,  
system and method for maintaining Web page state  
information even when the user moves from Web page to Web  
page. The present invention also provides an apparatus,  
system and method for storing Web page state information  
10 in a form that is secure and portable between a plurality  
of Web browser applications.

With the present invention, a user's request for a  
Web page is used to send a background request for Web  
page information from a directory server. The directory  
server stores Web page state information for every Web  
15 page visited by the user that requires a user's input.  
If the requested Web page information is stored in the  
directory server, the Web page information is downloaded  
to the user's client device and inserted into the  
received Web page at the appropriate fields in the Web  
20 page. This may be performed, for example, based on HTML  
tags, or the like.

If the Web page information is not stored in the  
directory server, the user's client device sends an  
instruction to the directory server to create a new Web  
25 page entry and to store the user's input into the Web  
page, in the new Web page entry of the directory server.

In another embodiment, the directory server stores a  
master directory listing for the user that includes  
information common to a plurality of Web pages. In  
30 addition, the directory server may also store individual  
Web page entries that contain information not already  
stored in the master directory listing. When a request  
for Web page information is received, the directory

09550181.041400

Docket No. AUS000078US1

server downloads both the master directory listing and the Web page entry to the requesting data processing system. The information is then inserted into the proper fields of the retrieved Web page based on, for example, HTML tags in the Web page document.

Other features and advantages of the present invention will be described in, or will be apparent to those of ordinary skill in the art in view of, the following description of the preferred embodiments.

10

09550181-044400



**BRIEF DESCRIPTION OF THE DRAWINGS**

5       The novel features believed characteristic of the  
invention are set forth in the appended claims. The  
invention itself, however, as well as a preferred mode of  
use, further objectives and advantages thereof, will best  
be understood by reference to the following detailed  
10 description of an illustrative embodiment when read in  
conjunction with the accompanying drawings, wherein:

**Figure 1** is an exemplary diagram of a typical Web  
page having fields for a user to enter personal  
information;

15       **Figure 2** is an exemplary diagram of a distributed  
data processing system in which the present invention may  
be implemented;

**Figure 3** is an exemplary diagram of a client device  
according to the present invention;

20       **Figure 4** is an exemplary block diagram of a directory  
server according to the present invention;

**Figure 5** is an exemplary diagram of a directory  
server Web page entry according to the present invention;

**Figure 6** is a flowchart outlining an exemplary  
25 operation of the directory server according to the present  
invention; and

**Figure 7** is a flowchart outlining an exemplary  
operation of the client device according to the present  
invention.

30

00550131-041400

**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

**Figure 2** is an exemplary diagram illustrating a distributed data processing system in which the present invention may be implemented. As shown in **Figure 2**, the system **200** includes a network **210**, at least one client device **220**, a plurality of Web page servers **230-250** and at least one directory server **260**.

The network **210** may be any type network that facilitates the transmission of data from one computing device to another. For example, the network **210** may be a Local Area Network (LAN), Wide Area Network (WAN), an Intranet, the Internet, or the like. Additionally, the network **210** may be any combination of the above networks. In the depicted example, distributed data processing system **200** is the Internet, with network **210** representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers consisting of thousands of commercial, government, education, and other computer systems that route data and messages.

The client device **220** sends and retrieves information to the Web page servers **230-250** via the network **210**. The client device **220** may be any type of computing device capable of transmitting and receiving data over the network **210**. For example, the client device **220** may be a personal computer, portable computer with a wired or wireless modem, Personal Digital Assistant (PDA), a Web TV™ device, Internet capable

09550181-041400

Docket No. AUS000078US1

telephone or pager, and the like. The client device 220  
retrieves Web pages from the Web page servers 230-250,  
sends data to the Web page servers 230-250, and sends and  
retrieves data to and from the directory server 260, as  
5 will be described hereafter.

The Web page servers 230-250 are computing devices  
configured to act as server devices in conjunction with  
network 210. The Web page servers 230-250 store Web  
pages that may be downloaded by the client device 220.  
10 Additionally, the Web page servers 230-250 may be capable  
of receiving data from the client device 220 for  
processing.

The directory server 260 may be any type of directory  
server that stores directory entries corresponding to the  
15 client device 220. For example, the directory server 260  
may be a Lightweight Directory Access Protocol (LDAP)  
server or Microsoft Active Directory Server.

The directory entries include information for each  
Web page downloaded by the client device 220 through  
20 which the client device 220 sent user information to the  
Web page server. The information stored in the directory  
entry may be used at a later time. For example, the  
directory entry information may be used when accessing  
the same Web page again to insert the previously entered  
25 user information into the appropriate fields of the Web  
page, or when accessing a different Web page that  
requires the information. In this way, the present  
invention eliminates the need for the user to repeat  
entry of the user information when the user has either  
30 moved to other Web pages or has not visited the Web page  
for some time.

In a preferred embodiment, the present invention is

0950131-041400

Docket No. AUS000078US1

implemented by an application running in the background of a Web browser used by the client device 220 to connect to the Web page servers 230-250 and download Web pages from the Web page servers 230-250. This application may  
5 be, for example, a plug-in application to a Web browser or may be a separate application that is run through multi-tasking in a multi-tasking environment such as Windows 98™ or the like.

When the client device 220 connects to a Web page  
10 server 230, for example, and sends a request to download a Web page, the identification of the requested Web page is also sent to the directory server 260 via the background application. The directory server 260 searches the directory entries for an entry matching the  
15 client device 220 and the requested Web page. If a matching entry is not found, the user of the client device 220 must enter the required information in the fields of the Web page as in the conventional sense.

With the present invention, however, the information  
20 entered by the user is not only sent to the Web page server 230 when the user sends a transmit command, but is also sent to the directory server 260 through the background application. The directory server then establishes a directory entry corresponding to the client  
25 device 220, the particular Web page, and the attributes and attribute values, i.e. the information entered by the user, as will be described in further detail hereafter.

Some information entered by the user may be personal information that is to be kept secret. In this case, the  
30 information may be encrypted and stored in an encrypted form (or BINARY attribute) in the directory server 260. Whether or not to encrypt information may be determined

09550181-041400

Docket No. AUS000078US1

based on a setting selected by the user through a Web browser, an HTML tag type, or the like. Information stored in the directory server 260 in an encrypted form may be retrieved and inserted into the Web page in either  
5 an encrypted or decrypted form.

If an entry in the directory server 260 already exists for the client device 220 and the Web page, the information in the entry is downloaded to the client device 220 and inserted in the appropriate fields of the  
10 Web page. The user need only send the transmit command to send this information to the Web page server 230. Alternatively, if the user wishes to modify the information in the entry, the user need only enter the new information in the appropriate fields over the  
15 information downloaded from the directory server 260. The modified information is then sent to the directory server 260 and stored as an entry in the directory server 260 in the same manner as for a new directory server 260 entry.

20 The insertion of the directory server 260 entry information into the Web page may be performed based on, for example, the HTML tags that are present in the Web page. For example, when a directory server 260 entry is created, the HTML tags and the associated values for the  
25 tags, i.e. the user's input into the fields of the Web page, are stored in correlation to one another in the directory server 260 entry. Thus, there may be a directory server 260 entry consisting of an HTML tag <socsec> that identifies a social security number tag. A  
30 corresponding input value is stored in the directory server 260 in association with the <socsec> tag.

When the directory server 260 entry information is

09550131-041400

Docket No. AUS000078US1

retrieved by the client device **220** for insertion into a retrieved Web page, the tags in the directory server **260** entry are matched to the tags in the HTML description of the Web page and the corresponding tag value is inserted  
5 into the associated field in the Web page. In this way, the user is relieved of the burden of having to repeatedly enter the same information each time he/she visits a Web page.

With reference now to **Figure 3**, a block diagram of a  
10 data processing system in which the present invention may be implemented is illustrated. Data processing system **300** is an example of a client computer, such as client device **220**. Data processing system **300** employs a Peripheral Component Interconnect (PCI) local bus  
15 architecture. Although the depicted example employs a PCI bus, other bus architectures, such as Micro Channel and ISA, may be used. Processor **302** and main memory **304** are connected to PCI local bus **306** through PCI bridge **308**. PCI bridge **308** may also include an integrated  
20 memory controller and cache memory for processor **302**. Additional connections to PCI local bus **306** may be made through direct component interconnection or through add-in boards. In the depicted example, Local Area Network (LAN) adapter **310**, SCSI host bus adapter **312**, and  
25 expansion bus interface **314** are connected to PCI local bus **306** by direct component connection. In contrast, audio adapter **316**, graphics adapter **318**, and audio/video adapter (A/V) **319** are connected to PCI local bus **306** by add-in boards inserted into expansion slots. Expansion  
30 bus interface **314** provides a connection for a keyboard and mouse adapter **320**, modem **322**, and additional memory **324**. In the depicted example, SCSI host bus adapter **312**

09550131-041400

09550131 "041400

provides a connection for hard disk drive **326**, tape drive **328**, CD-ROM drive **330**, and Digital Video Disc Read Only Memory drive (DVD-ROM) **332**. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors. An operating system runs on processor **302** and is used to coordinate and provide control of various components within data processing system **300** in **Figure 3**. The operating system may be a commercially available operating system, such as OS/2, which is available from International Business Machines Corporation. "OS/2" is a trademark of International Business Machines Corporation. An object oriented programming system, such as Java, may run in conjunction with the operating system, providing calls to the operating system from Java programs or applications executing on data processing system **300**. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on a storage device, such as hard disk drive **326**, and may be loaded into main memory **304** for execution by processor **302**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 3** may vary depending on the implementation. For example, other peripheral devices, such as optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 3**. The depicted example is not meant to imply architectural limitations with respect to the present invention. For example, the processes of the present invention may be applied to multiprocessor data processing systems.

The data processing system **300** may communicate with

one or more Web page servers, such as Web page servers 230-250, via the LAN adapter 310, the modem 322, or the like. The data processing system utilizes a Web browser program executed by the processor 302, as is generally  
5 known in the art, to enter URLs of Web sites and thereby download Web pages from Web page servers.

In addition, the processor 302 executes a background application for communication with a directory server, such as directory server 260. The background application  
10 establishes a communication connection with the directory server using, for example, TCP/IP. When a Web page is downloaded from a Web page server, the background application sends a request for directory entry information corresponding to the Web page identifier and  
15 the data processing system identification.

In response to receiving a request for directory entry information, the directory server searches the directory listings for an entry matching the particular data processing system identification and the Web page  
20 identifier. If an entry is present, the directory entry information is downloaded to the data processing system 300. If an entry is not present, a new entry is created and user inputs into fields of the Web page are transmitted to the directory server 260 via the  
25 background application.

**Figure 4** is an exemplary block diagram of a directory server 400 according to the present invention. As shown in **Figure 4**, the directory server 400 includes a controller 410, a network interface 420 and a Web page  
30 entry storage device 430. These devices are in communication with one another via the control/signal bus 440. Although a bus architecture is shown in Figure 4,

0950181-044400



the invention is not limited to such an architecture. Rather, any architecture that facilitates communication between the elements 410-430 may be used without departing from the spirit and scope of the present invention.

When the controller 410 receives a request for directory information via the network interface 420, the controller initiates a search of the entries in the Web page entry storage device 430. The search is performed based on the requesting data processing system's identification and the Web page identification. If a matching entry is found in the Web page entry storage device 430, the entry information is downloaded to the requesting data processing system via the network interface 420. If a matching entry is not found, the controller 410 sends a "not found" message to the requesting data processing system.

In response to receiving the "not found" message, the background application of the requesting data processing system collects the HTML tags and the corresponding user's input into the Web page and sends this information to the directory server 400 along with an instruction to create a new Web page entry. When the controller 410 receives the instruction from the requesting data processing system, the controller 410 creates a new entry in the Web page entry storage device 430 based on the requesting data processing system's identifier and the Web page identifier. The controller 410 then stores the HTML tags and the corresponding user input into the Web page entry storage device 430 in association with the created Web page entry.

Figure 5 is an exemplary diagram illustrating a Web

0950181-04400

Docket No. AUS000078US1

page entry in a directory server. As shown in **Figure 5**, the Web page entry includes a plurality of attributes 510 and a plurality of corresponding values 520. The attributes include a user ID, or a data processing system identifier, a page name (pn), a page number (no.), an application name (an), a social security number (socsec), a date of birth (dob), an electronic mail address (email), and the like. The attributes have corresponding values 520 of "tran@lptr.com", "www.ibm.com", "2", "register.html", "999-46-5699", "021775", and "tran@ges.com". Other attributes may be included in the Web page entry either in addition to or in place of the attributes shown in **Figure 5**.

When the directory server performs a search to determine if there is a Web page entry corresponding to requested directory information, the directory server first searches for a user ID corresponding to the identification of the requesting data processing system. If a matching user ID is found, the directory server then searches for those entries matching the user ID that also match the requested Web page name (pn). If a matching entry is found, the corresponding attributes and values are downloaded to the requesting data processing system.

If a corresponding entry is not found during the directory server's search, a new entry is created using the requesting data processing system's identifier as a user ID, and the URL of the requested Web page as the page name (pn), for example. Thereafter, user input to the Web page is received and stored as attribute/value pairs in the Web page entry.

**Figure 6** is a flowchart outlining an exemplary operation of the present invention from the point of view

09550131-041400

Docket No. AUS000078US1

of the directory server. As shown in **Figure 6**, the operation starts when a request for Web page information is received from a requesting data processing system (step **610**). A search of the directory entries for the requested Web page information is conducted (step **620**). A determination is made as to whether a matching Web page entry is found in the directory of entries (step **630**). If a matching Web page entry is found in the directory (step **630:YES**), the Web page information is downloaded to the requesting data processing system (step **650**).

With reference again to step **630**, if a matching Web page entry is not found in the directory (step **630:NO**), a new Web page entry is created using the identification information for the requesting data processing system and the requested Web page, along with the user's input into the Web page (step **640**). Thereafter, a determination is made as to whether there are additional Web pages (step **660**). If there are more Web pages (step **660:YES**), i.e. the user moves to another Web page using a Web browser, the operation returns to step **610** and repeats the operation. Otherwise, the operation ends.

**Figure 7** is a flowchart outlining an exemplary operation of the present invention from the point of view of the client device. As shown in **Figure 7**, the operation starts with the entry of a URL or other type Web page identifier into a Web browser (step **710**). Thereafter, a background request for Web page information is sent to a directory server (step **720**).

A determination is made as to whether the requested information is received (step **730**). If the requested information is received (step **730:YES**), the received information is inserted into the appropriate fields of

09550131.04.14.00

the requested Web page. If the requested information is not received (step 730:NO), an instruction is sent to the directory server to create a new Web page entry (step 750). Thereafter, the user's input to the requested Web page is also sent to the directory server and stored in the Web page entry (step 760).

Then, a determination is made as to whether there are additional Web pages (step 770). If there are more Web pages (step 770:YES), the operation is continued at step 710. Otherwise, the operation ends.

As discussed above, the present invention may be embodied as a plug-in application to a Web browser or as a separate background application. In one particular embodiment, the present invention may be either a plug-in application or a background application written in the Java programming language. With such an application, two JavaScripts called GetState(Attr) and SetState(Attr, Value) are provided. The SetState(Attr, Value) JavaScript allows the application to store the Web page entry by storing pairs of attributes and corresponding values. The GetState(Attr) JavaScript allows the application to retrieve the attribute values from the directory server in which they are stored accordingly.

Thus, with the present invention, Web page state information is stored in a directory server for use when later accessing a Web page again. In this way, Web page state information is not lost when the user decides to move from Web page to Web page. Furthermore, the user is freed from having to repeatedly input the same information each time he/she accesses a particular Web page. In addition, the information stored in the directory server may be encrypted for greater security.

0950431-044400

Docket No. AUS000078US1

Because all Web browser programs are HTML compatible and the attribute and value information are stored based on HTML tags, the Web page entry is compatible with all Web browsers regardless of the particular Web browser used  
5 when the Web page entry was created.

In another embodiment of the present invention, the user's personal information is not stored for each individual Web page visited by the user. Rather, a master directory listing is provided that contains  
10 attributes and associated values that are commonly required by a plurality of Web pages. This master directory listing may include such information as the user's name, address, date of birth, age, marital status, and the like.

15 The master directory listing may be stored in place of or in addition to individual Web page directory listings. The individual Web page directory listings include all information requested by the Web page that is not already stored in the master directory listing. In  
20 this way, when a Web page directory listing is downloaded to a requesting data processing system, the master directory listing is also downloaded. Proper insertion of the various information is achieved by matching HTML tags to insert the appropriate information. Thus,  
25 storage space requirements of the directory server are reduced.

Although the above embodiments are directed to the use of HTML Web pages and the use of the World Wide Web, the invention is not limited to such embodiments.  
30 Rather, the present invention is applicable to other Internet resources for transferring information, such as File Transfer Protocol (FTP) and Gopher.

It is important to note that while the present

00441400 00550131

invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, CD-ROMs, and transmission-type media such as digital and analog communications links.

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

00440-EST-0550